

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

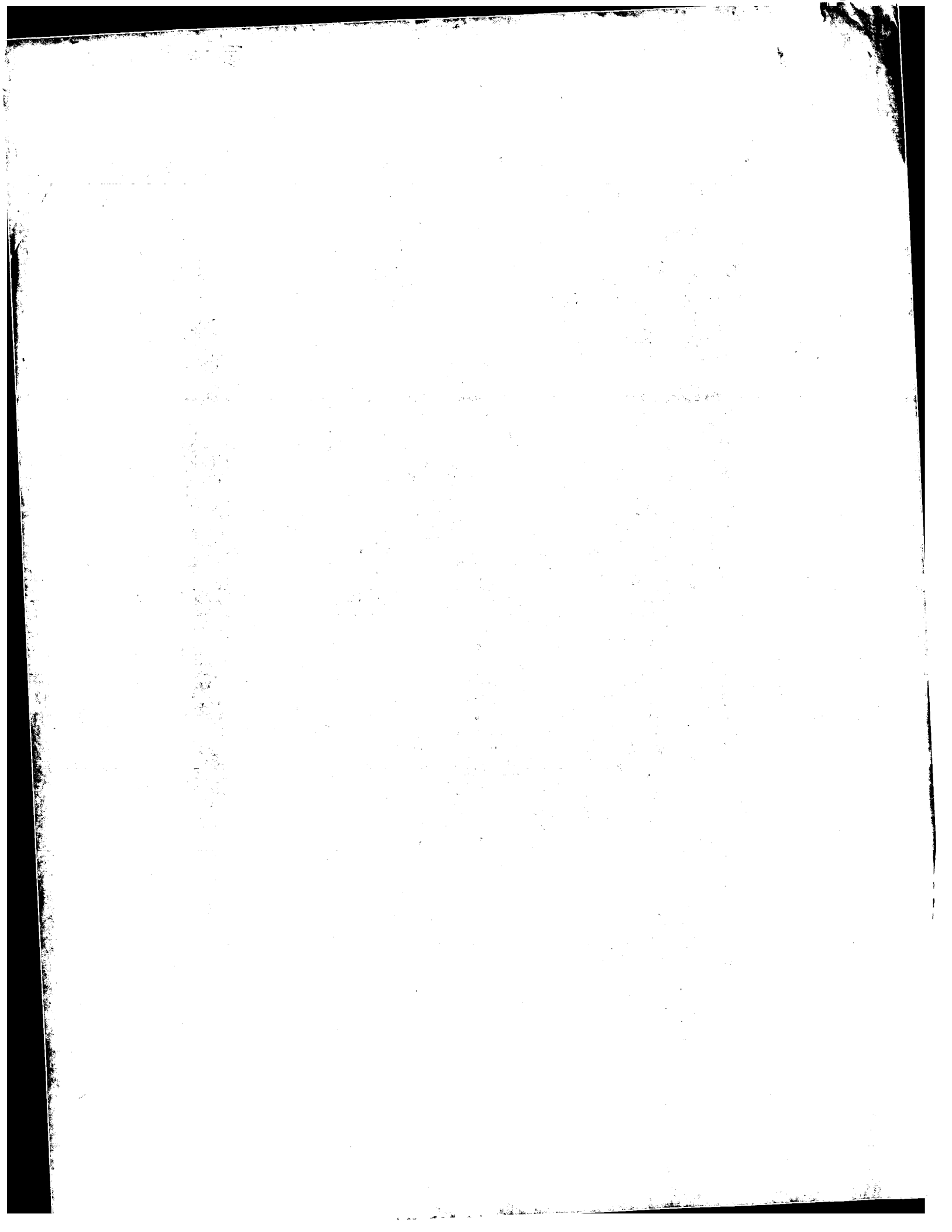
- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Problem Image Mailbox.**

Japanese Reference X
(Statement of Relevance)

This reference was cited during an opposition proceeding for a Japanese patent (JP 2995156 granted 11/22/99) issuing from an application corresponding to U.S. patent application Serial No. 07/475,111.



著者紹介

上坂吉則

1959年 名古屋工業大学電気工学科卒業
NHK放送科学基礎研究所を経て
1973年 金沢大学農医(工学部)
1979年 東京理科大学農医(理工学部)
現在に至る。工学博士
主著書籍「パターン認識と学習の理論」(文芸春秋出版、1971)
「情報科学の基礎」(朝倉書店、1977)
「情報科学の基礎」(朝倉書店、1979)

太原育夫

1974年 東京大学工学部電子工学科卒業
1979年 東京大学工学部電子工学科助手(博士課程修了)
工学博士
東京理科大学農医(理工学部)
1981年 東京理科大学農医(理工学部)
現在に至る。

上坂吉則 著
太原育夫

パソコンで学ぶ

パターン認識と図形処理

パソコンで学ぶ
パターン認識と図形処理

上坂吉則 著
太原育夫

検印
省時

1984年11月30日 第1刷発行
1985年12月20日 第2刷発行

発行所

株式会社 文一総合出版

代表者 奥村 武

〒102 東京都港区五軒町34-8
電話 03-235-7241~3

奥村印刷：製本

© 1984

Printed in JAPAN

ISBN 4-8299-2068-8

株式会社文一総合出版

11 パターン・マッチング——DPマッチング

テンプレートの変形をある程度許容しながら入力パターンとのマッチングを行なう方法を考える。ミスマッチの測度を計算する過程で動的計画法が用いられる。

説明

パターン・マッチング(整合)は、あらかじめ用意されたテンプレート(参照パターン)を入力パターンに重ね合せ、それらが一致するかどうかに基づいて入力パターンの識別を行なう最も基本的なパターン認識の手法である。しかし、原理は簡単であっても、マッチングの方法やマッチ(あるいは、ミスマッチ)の度合の評価にはいろいろなものが考えられる。

1では、入力パターンに対し、テンプレートをそのまま重ね合わせてマッチングを行なう最も簡単なマッチング法について述べた。この章では、テンプレートをある制限のもとで変形させながら入力パターンと重ね合わせ、最もよくマッチしたときのずれの量(パターン間の距離)をミスマッチの測度とするようなマッチング法を考える。

テンプレートを F 、入力パターンを G として N 次元の時系列パターンを考えることにする：

$$F: F(i) = (F_1(i), F_2(i), \dots, F_M(i)) \quad (0 \leq i \leq T_F),$$

$$G: G(i) = (G_1(i), G_2(i), \dots, G_M(i)) \quad (0 \leq i \leq T_G).$$

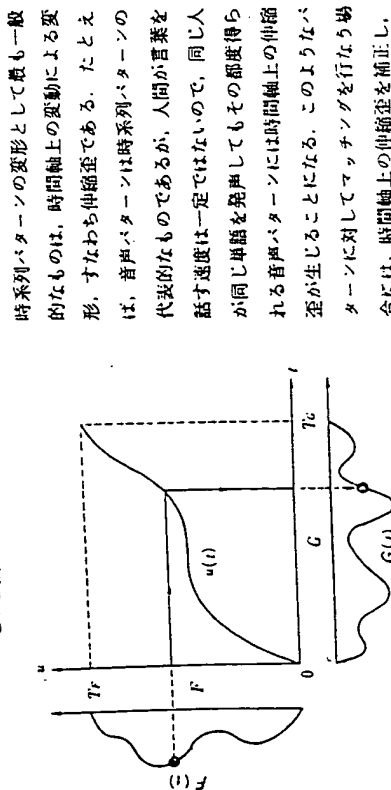


図 11.1 関数 $u(i)$ による時間軸の伸縮変換

似た状態になるようにしてから両者の間のずれを計量すべきであると考えられる。そこで、テンプレート F の時系列関数 $u(i)$ を用いて伸縮交換することにする。 $u(i)$ は図11.1に示すよ

うな連続な単調増加関数で

$$u(0) = 0,$$

$$u(T_G) = T_F$$

である。このように時間軸を変換したテンプレート $F(u(i))$ と入力パターン $G(i)$ の間の距離 $E(F, G, u(i))$ を

$$E(F, G, u(i)) = \frac{\|G(i) - F(u(i))\|}{\int ds} \quad (11.1)$$

と定義することにより、ここで

$$\|G(i) - F(u(i))\| = \sqrt{\sum_{m=1}^M (G_m(i) - F_m(u(i)))^2},$$

$$ds = \sqrt{1 + \left(\frac{du}{di}\right)^2} di$$

であり、積分は $0 \leq i \leq T_G$ について行なうものとする。この $E(F, G, u(i))$ を最小にするような $u(i)$ を選択すれば、 F と G が最も近似した状態になるはずである。したがって、そのときの $E(F, G, u(i))$ の値を F と G のミスマッチの測度 $D(F, G)$ とすればよい。

$$D(F, G) = \min_{u(i)} [E(F, G, u(i))] \quad (11.2)$$

さて、テンプレート F 、入力パターン G がともに時間軸上で標準化された時系列パターンである場合について考えることにしよう：

$$F = f_1, f_2, \dots, f_p, \dots, f_n,$$

$$G = g_1, g_2, \dots, g_n, \dots, g_n,$$

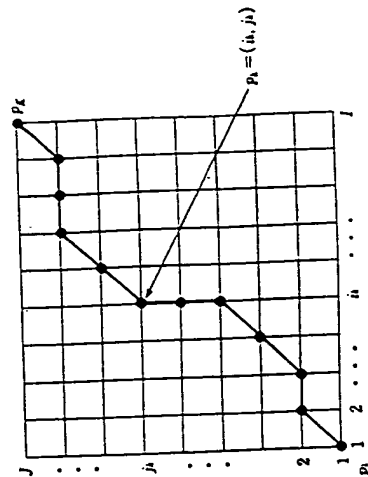


図 11.2 格子グラフ

この場合、時間軸の伸縮変換 $u(i)$ は、図11.2に示すような格子グラフ上の点 $(1,1)$ から点 (J,J) へ至るルートによって表現される。したがって、マッチングにおける F の標準点と G の標準点の対応関係 P はこのルート上の点列で与えられることになる。格子点を

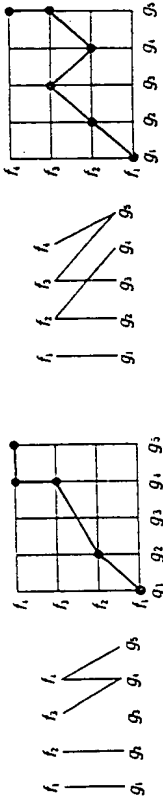
$$p_k = (i_k, j_k) \quad (1 \leq i_k \leq J, 1 \leq j_k \leq J)$$

とすれば

$$P = p_1, p_2, \dots, p_K$$

である。時間軸の伸縮変換が不自然なものにならないようにするため、ルートの選択には少なくともつぎのような制限が加えられる：

- (1) 連続性と単調増加性、
- (2) $p_1 = (1, 1)$ かつ $p_K = (I, J)$ 。



(a) 連続性に反する例

(b) 単調増加性に反する例

図 11.3 連続性と単調増加性に反する例

したがって、図11.3に示すようなルートの選択は許されないことになる。条件(1)を満足するようなルートの選択方法はいくつ考えられるが、ここでは最も一般的な

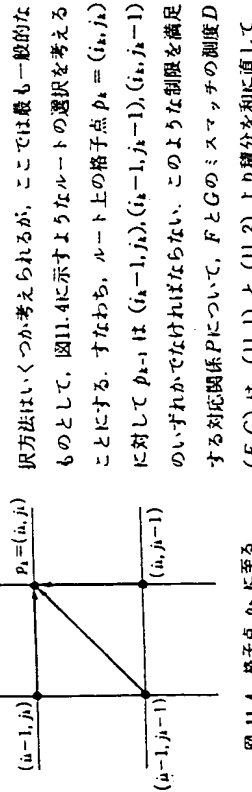


図 11.4 格子点 p_k に至るルート

$$D(F, G) = \min_P \left[\frac{\sum_{k=1}^K d(p_k) d_k}{\sum_{k=1}^K d_k} \right] \quad (11.3)$$

で与えられる。ここで

$$d(p_k) = \|g_{i_k} - f_{j_k}\|$$

であり、 d_k は (11.1) の d_s に対応している。

具体的に与えられた F と G について $D(F, G)$ の値を求める場合、(11.3) をともにそのまま計算したのでは、最小値を得るのに相当な量の計算が必要となる。一般に、このような最小化問題は、動的計画法によって少なくて済む計算で効率よく解くことができる。 d_k を

$$d_k = (i_k - i_{k-1}) + (j_k - j_{k-1}),$$

ただし

$$i_0 = j_0 = 0$$

と定義すれば

$$\sum_{k=1}^K d_k = I + J$$

であるから、(11.3) は

$$D(F, G) = \frac{1}{I+J} \min_P \left[\sum_{k=1}^K d(p_k) d_k \right] \quad (11.4)$$

となり動的計画法が適用できる。

標本点間の対応関係 P に関して、つぎのような部分和を考える：

$$h(p_k) = \min_{p_1, p_2, \dots, p_{k-1}} [d(p_1) d_1 + d(p_2) d_2 + \dots + d(p_k) d_k] \quad (k \leq 2) \quad (11.5)$$

この部分 (11.5) はただちに漸化式の形に変形でき

$$h(p_k) = \min_{p_{k-1}} [h(p_{k-1}) + d(p_k) d_k] \quad (11.6)$$

となる。ここで $h(p_1) = d(p_1) d_1 = 2d(p_1)$ である。ところで、格子グラフ上でのルートの選択には図11.4のような制限があるので、(11.6) は

$$h(p_k) = \min \left\{ \begin{array}{l} h(i_k - 1, j_k - 1) + 2d(i_k, j_k) \\ h(i_k - 1, j_k) + d(i_k, j_k) \\ h(i_k, j_k - 1) + d(i_k, j_k) \end{array} \right\} \quad (11.7)$$

となる。この漸化式により、 $h(p_k)$ は $h(p_1)$ を初期条件として順次求めることになる。このようにして $h(p_K)$ が求まれば、ミスマッチの測度 $D(F, G)$ は

$$D(F, G) = \frac{1}{I+J} h(p_K) \quad (11.8)$$

となる。

以上のような方法によるパターン・マッチングは、ミスマッチの測度を計算する過程で動的計画法 (Dynamic Programming) が用いられるので、DP マッチングと一般に呼ばれている。

パターン・マッチング プログラム

テンプレート F 、入力パターン G として1次元の時系列パターンが与えられたとき、DP マッチングにより、 F と G のミスマッチの測度 $D(F, G)$ を求めるプログラムを考える。

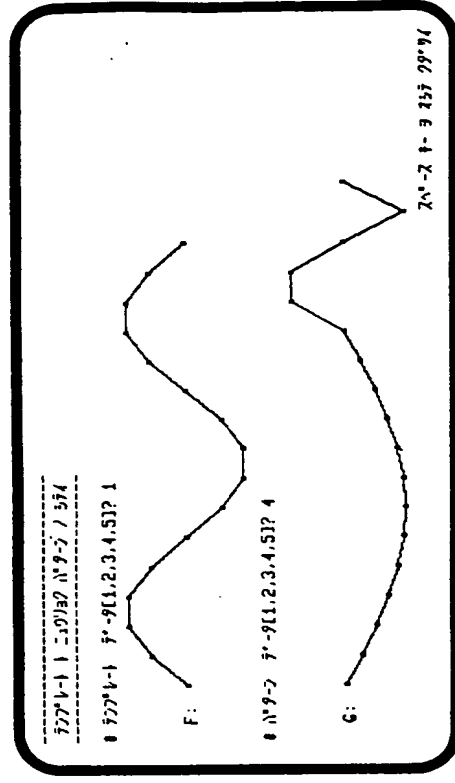
一般に、DP マッチングを用いて入力パターンの認識を行なう場合には、単に $D(F, G)$ の値が求まりさえすればよく、そのときの対応関係 P は知る必要がないので、漸化式 (11.7) をそのまま計算するだけでよい。しかし、本章の目的は DP マッチングそのものを理解することにあるので、このプログラムでは、 $D(F, G)$ を与える対応関係 P がどのようなものになっているのかという点とも表示することにする。対応関係 P を求めるには、配列 $P(i_k, j_k)$ ($1 \leq i_k \leq I, 1 \leq j_k \leq J$) を用意しておき、漸化式 (11.7) の計算において最小値を与える格子点 p_{k-1} の p_k に対する相対位置をその郵便記憶しておけばよい。すなわち、 $h(p_k)$ を求めたとき最小値を与える p_{k-1} が

$$p_{k-1} = (i_k - 1, j_k - 1) \text{ ならば } P(i_k, j_k) = 1,$$

X7, X8, X9, Y7, Y8, Y9, X7%, X8%, Y7%, Y8% = グラフィック・サブルーチン用の
入力変数 (付録A参照)。

実行結果

プログラムの実行例を以下に示す。プログラムを実行すると、まず表題が表示される。スペース・キーを押すと、プログラム末尾にある5つの時系列データのうちのどれをテンプレートFとし、どれを入力パターンGとするか指定するよう要求してくる。ここでは、例としてデータ番号1の時系列データをテンプレートとし、データ番号4の時系列データを入力パターンとすることにする。データ番号を入力すると、それぞれのデータが読み込まれ表示される。



指示に従いスペース・キーを押すと、DP マッチングの計算が開始される。計算が終了し、FとGのミスマッチの測定 $D(F, G)$ の値とそれを与える標本点間の対応関係 P が求まると、まず対応関係 P が格子グラフによって表示される。ディスプレイ上の格子グラフにおいて、縦軸はテンプレートFの標本点、横軸は入力パターンGの標本点をそれぞれ表わしている。

11 パターン・マッチング—DP マッチング

$$p_{k-1} = (i_k - 1, j_k) \text{ ならば } P(i_k, j_k) = 2,$$

$$p_{k-1} = (i_k, j_k - 1) \text{ ならば } P(i_k, j_k) = 3$$

としておけばよい。DP マッチングの終了後、 $i_k = I, j_k = J$ を初期条件として、順次

$$\left. \begin{aligned} P(i_k, j_k) &= 1 \text{ ならば } i_{k-1} = i_k - 1, \quad j_{k-1} = j_k - 1 \\ P(i_k, j_k) &= 2 \text{ ならば } i_{k-1} = i_k - 1, \quad j_{k-1} = j_k \\ P(i_k, j_k) &= 3 \text{ ならば } i_{k-1} = i_k, \quad j_{k-1} = j_k - 1 \end{aligned} \right\} \quad (11.9)$$

とすることにより、格子点 $p_k = (i_k, j_k)$ ($1 \leq k \leq K$) を求めれば、 $D(F, G)$ を与える対応関係 P が得られる。

プログラムの全体の流れはつぎのようになっている：

1° 表題の表示。

2° 初期設定。

3° テンプレートと入力パターンの指定。

指定されたデータ番号により、プログラム末尾にある時系列パターンのデータのなかからテンプレートFとするパターン、入力パターンGとするパターンを読み込む。読み込んだデータはただちにディスプレイ上に表示する。

4° DP マッチングの計算。

漸化式 (11.7) を用いてミスマッチの測定 $D(F, G)$ を計算する。また、 $D(F, G)$ を与える対応関係 P を求めるため、この計算と平行して $h(p_k)$ を与える格子点 p_{k-1} の相対位置を配列 $P(i_k, j_k)$ に記憶しておく。

5° 対応関係の表示。

ミスマッチの測定 $D(F, G)$ を与える対応関係 P を (11.9) に従って求め、格子グラフ上のルートとして表示する。

6° 結果の表示。

テンプレートF、入力パターンGの時系列パターンの標本点が具体的にどのような対応付けられたのが表示し、かつ計算の結果得られたミスマッチの測定 $D(F, G)$ の値を表示する。

なお、このプログラムではグラフィック表示を用いるので、実行の際には付録Aのグラフィック・サブルーチンを実行していただきたい。

プログラムで用いている主な変数はつぎのとおりである：

F = 大きさ NF の1次元配列でテンプレートF用の時系列パターン。

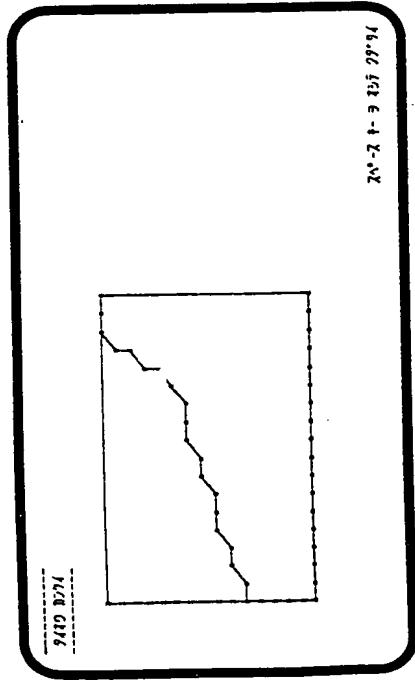
NF = テンプレートFの標本数。

G = 大きさ NG の1次元配列で入力パターンG用の時系列パターン。

NG = 入力パターンGの標本数。

H = (11.5) で定義される $h(i_k, j_k)$ を表わす。

P = 対応関係 P を求めるための配列 $P(i_k, j_k)$ を表わす。



スペース・キーを押すと、FとGの標本点が具体的にどのように対応付けられているのか表示される。そして、そのような対応関係における $D(F, G)$ の値が表示される。 $D(F, G)$ はミスマッチの測度であるから、FとGが同一のパターンであればもちろん0となり、異なれば異なるほど大きな値となる。プログラムの末尾にある5つの時系列パターンデータのデータについて比較検討してみるとよい。

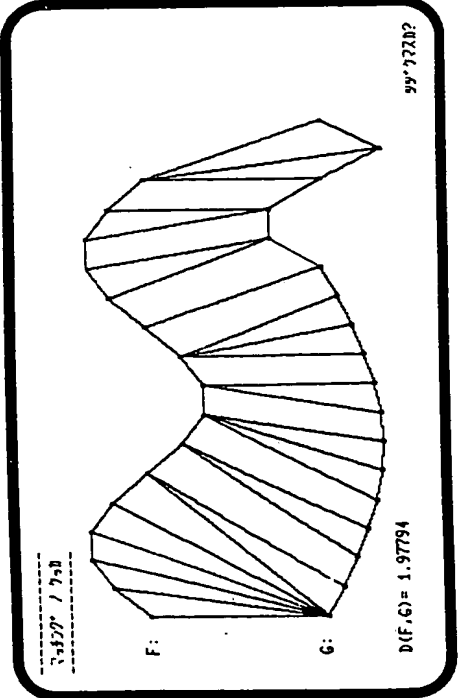


図 11.5 スプリング負荷モデルを用いたパターン・マッチング

DP マッチングは、時間軸上の伸縮歪をもつ時系列パターン間のマッチングだけでなく、2次元平面上の図形パターンや画像間のマッチングにも適用することができる。その場合、主として2通りの適用のしかたが考えられる。

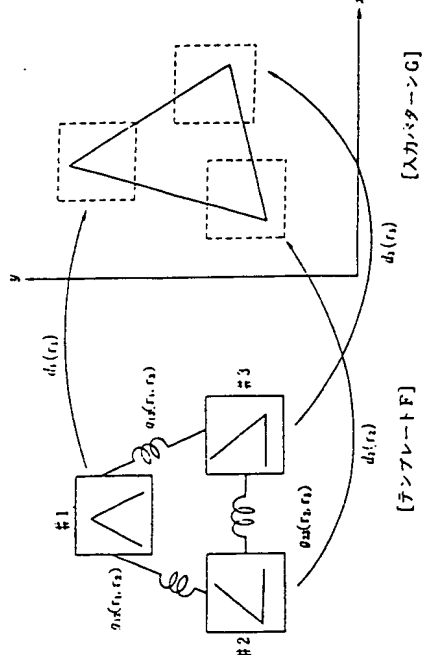


図 11.6 2次元平面上の線図形パターンをフリーマンのチェーン・コード (17 参照) や全曲率関数 (18 参照) などを用いて1次元の記号列あるいは1変数の関数で表現する方法である。このような表現をすれば、2次元平面上的パターンも時系列パターンと同様な取扱いが可能になり、DP マッチングをそのまま適用することができる。

もう1つの方法は、テンプレートの変形のしかたを2次元に拡張する方法である。この場合、一般にテンプレートとしてはスプリング負荷モデルが用いられる。スプリング負荷モデルは、図 11.5 に示すように、いくつかの部分パターンをスプリングで結合した構造をしている。このようなテンプレートを用いてマッチングを行なう場合には、スプリング負荷モデルのスプリングを伸ばしながら入力パターンに重ね合わせ、最もよくマッチしたとき、すなわち各部分パターン自体がそれぞれよくマッチし、かつスプリングにかかる負荷が全体としてできるだけ少ないときの量を求め、それをミスマッチの測度とすればよい。

いま、テンプレートFとしてN個の部分パターン ($\#1 \sim \#N$) からなるスプリング負荷モデルを考え、入力パターンGが定義されている2次元平面に重ね合わせるものとする(図 11.5 参照)。部分パターン $\#i$ の位置を $r_i = (x_i, y_i)$ としたとき、部分パターン自体のミスマッチの測度を $d_i(r_i)$ とする。また部分パターン $\#i$ が $r_i = (x_i, y_i)$ にあり、部分パターン $\#j$ が $r_j = (x_j, y_j)$ にあったとき、両者を結合しているスプリングにかかる負荷を $g_{ij}(r_i, r_j)$ とする。この $g_{ij}(r_i, r_j)$ としては、たとえば $k_{ij} l_{ij}$ を定数として

$$g_{ij}(r_i, r_j) = k_{ij}(|r_i - r_j| - l_{ij})$$

というような関数を考えればよい。さて、N個の部分パターンの配置を R_N としたとき、部分パターン $\#i$ から $\#j$ の配置を R_{ij} で表現することにする：

$$R_{ij} = \{r_{i1}, r_{i2}, \dots, r_{ij}\}$$

このとき、配置 R_{ij} に関する部分パターン $\#i$ のずれの量 $h_i(R_{ij})$ を

$$(11.10)$$

